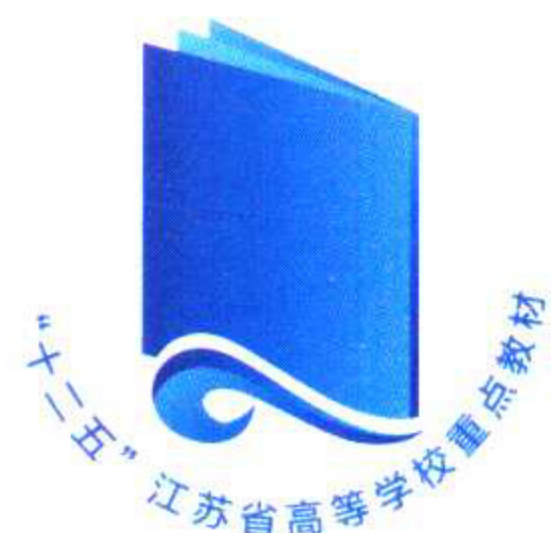




“十二五”普通高等教育本科国家级规划教材

“十二五”江苏省高等学校重点教材

新工科建设之路·计算机类专业规划教材



Java 程序设计实用教程

第 5 版

叶核亚 编著 陈道蓄 主审



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



Java 程序设计实用教程

第 **5** 版

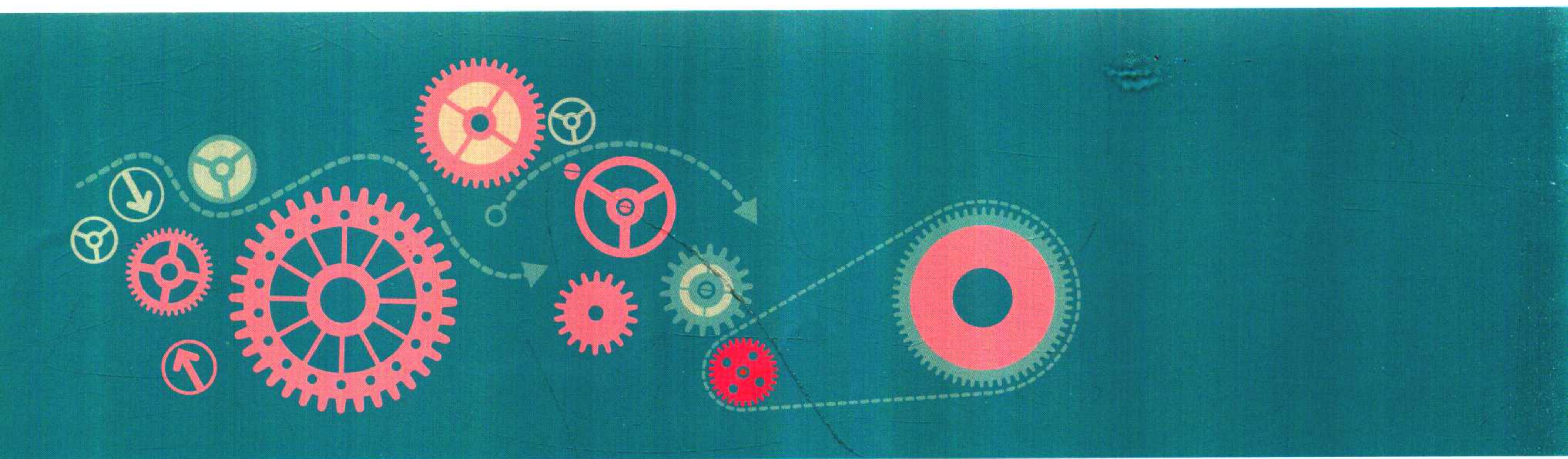
本书是“十二五”普通高等教育本科国家级规划教材。

本书全面介绍 Java 语言程序设计的基础知识、运行机制、多种编程方法和技术，力求建立牢固扎实的理论基础，系统、有序地进行程序设计和面向对象方法的基础训练；为操作系统、数据库应用、网络通信、Web 应用等软件领域的实际应用问题，提供基于 Java 技术的解决方案。

本书主要内容包括：Java 开发运行环境 JDK 和 MyEclipse，Java 语言基础，类的封装、继承和多态，接口、内部类和 Java API，异常处理，图形用户界面，多线程，输入/输出流和文件操作，网络通信，数据库应用，Web 应用和综合应用设计。这些内容是构成 Java 应用程序的基本要素和必备知识。

本书注重理论与实践相结合，注重基础知识的理解与基本技能的培养。全书内容丰富，结构安排合理，由浅入深，层次分明，内容涉及的广度和深度符合本科培养目标的要求。

本书可作为普通高等学校计算机及相关专业本科的 Java 语言程序设计课程教材，也可作为使用 Java 语言从事软件开发人员的参考书。



策划编辑：章海涛
 责任编辑：章海涛
 封面设计：欧美尼 李根星



定价：58.00 元

“十二五”普通高等教育本科国家级规划教材

“十二五”江苏省高等学校重点教材

新工科建设之路·计算机类专业规划教材

Java 程序设计实用教程

第 5 版

叶核亚 编著
陈道蓄 主审



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是“十二五”普通高等教育本科国家级规划教材。

本书全面介绍 Java 语言程序设计的基础知识、运行机制、多种编程方法和技术，力求建立牢固扎实的理论基础，系统、有序地进行程序设计和面向对象方法的基础训练；为操作系统、数据库应用、网络通信、Web 应用等软件领域的实际应用问题，提供基于 Java 技术的解决方案。

本书主要内容包括：Java 开发运行环境 JDK 和 MyEclipse，Java 语言基础，类的封装、继承和多态，接口、内部类和 Java API，异常处理，图形用户界面，多线程，输入/输出流和文件操作，网络通信，数据库应用，Web 应用和综合应用设计。这些内容是构成 Java 应用程序的基本要素和必备知识。

本书可作为普通高等学校计算机及相关专业本科的 Java 语言程序设计课程教材，或使用 Java 语言从事软件开发人员的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

Java 程序设计实用教程 / 叶核亚编著. —5 版. —北京: 电子工业出版社, 2019.1

ISBN 978-7-121-34441-1

I. ① J… II. ① 叶… III. ① JAVA 语言—程序设计—高等学校—教材 IV. ① TP312.8

中国版本图书馆 CIP 数据核字 (2018) 第 119813 号

策划编辑: 章海涛

责任编辑: 章海涛

特约编辑: 何 雄

印 刷: 北京京师印务有限公司

装 订: 北京京师印务有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1092 1/16 印张: 27 字数: 690 千字

版 次: 2003 年 3 月第 1 版

2019 年 1 月第 5 版

印 次: 2019 年 1 月第 1 次印刷

定 价: 58.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：192910558 (QQ 群)。

前言

软件设计的思想和方法以及所采用的程序设计语言,都必须跟随软件时代的发展而不断改进和更新。面向对象程序设计方法是目前软件开发的主流方法。Java 语言是目前功能最强、应用最广泛的一种完全面向对象程序设计语言,具有成熟而严密的语法体系、跨平台特点和强大的应用系统设计能力。今日 Java 应用无处不在,Java 作为程序设计的首选语言,其重要性毋庸置疑。因此,采用 Java 语言进行面向对象的程序设计方法训练是十分恰当的,是程序设计系列课程教学改革的必然,完全符合本科培养目标的要求。

开设 Java 程序设计课程的目的:一是采用 Java 语言进行程序设计和面向对象方法的基础训练,二是运用操作系统中的线程、文件概念,网络原理,数据库原理等基础理论进行线程、文件、网络、数据库、Web 等应用的设计训练。

Java 技术不仅能够实现这些功能,还可以使算法表达更简明、更直接,性能更好。

本书是“十二五”普通高等教育本科国家级规划教材,可作为普通高等学校计算机及相关专业本科生的 Java 语言程序设计课程教材。

本书力求展现“理论基础厚实、全面展现应用技术、加强工程应用能力培养”的鲜明特色,不仅全面阐述面向对象概念,还通过各种应用实例展示 Java 技术,充分体现 Java 语言的优越性,让学生看见、体会并深刻理解,再通过强化实践环节等措施加强工程应用能力的培养,努力探索出一套适合工科院校计算机类专业的教学方案,体现 Java 作为专业主干课程的重要作用。这也是新工科建设的落实。

(1) 理论基础厚实

本书全面、系统地介绍构成 Java 应用程序的基本要素和必备知识,包括 Java 跨平台的运行机制、Java 语言的语法基础和面向对象基本概念,着重介绍类的封装、继承和多态等面向对象的核心特性,以及接口、内部类、包、异常处理等 Java 特有的实现机制。

全书结构安排合理,由浅入深,层次分明,章节之间有机衔接,前后呼应,内容涉及的广度和深度符合本科培养目标的要求,配套教学资源齐全。

(2) 全面展现 Java 应用技术

本书介绍 Java 语言的图形用户界面、多线程、输入/输出流和文件操作、网络通信、数据库应用、Web 应用等应用技术。这些知识和技术与现实世界联系紧密,实用性较强,学生易于理解,但实现起来较困难。本书以面向对象理论为基础,以广阔的实际应用为背景,采用一个个贴近生活实际的实例展现 Java 应用技术,展现面向对象思想的作用和使用方法,说明为什么 Java 能够在网络环境中被广泛应用,Java 的哪些机制使其具有独特的魅力,从而能够更深入地理解面向对象思想的精妙。

(3) 加强工程应用能力培养

“Java 程序设计”是一门理论与实践并重的课程,不仅要理解基础知识,更要培养软件设计的基本技能。实践性环节是巩固所学理论知识、积累程序设计经验的必不可少的重要环节,

是提高程序设计能力和计算机操作技能的有力保障。

注重理论基础和实用技术相结合，注重在实践环节培养程序设计的基本技能，是本书的重要特色。本书将程序设计能力的锻炼和提高设计为一个循序渐进的过程，将基本原理体现在从原理叙述、例题、思考题等课堂讲授环节，到课后习题、上机实验、课程设计等实践性环节，让学生经历“先见识、再模仿、最后自主创新设计”的学习过程，并培养良好的程序设计习惯。

每章安排有习题和上机实验题，给出详细的实验训练目标、设计内容和设计要求。实验题精心选择，形式多样，生动有趣，引人入胜，难度逐步增加。

针对课程设计的实践性环节，本书给出了综合应用程序设计实例，详细说明了需求方案、设计目标、设计任务、模块划分、功能实现、调试运行等环节的设计方法，贯彻了理论讲授和案例教学相结合的教学方法，既训练学生具有扎实深厚的基本功，也具有可扩展素质和很强的创新能力。

本书采用的运行环境有 JDK 8、MyEclipse 2015、MySQL 5.7 数据库和 Tomcat 8.0。

这是一本写程序设计的书。程序设计有一些基本原则和道理。程序是设计出来的，程序员必须具备基础知识和基本技能，在写程序之前需要根据实际应用需求，从全局角度通盘规划考虑，精心策划，选择采取什么策略，清楚有哪些方法可以做得成，每种方法都有什么优缺点，明白为什么要这样做，那样做为什么就不可以，等等。

程序中发现错误了怎么办？这是什么错误，是否能改正，如何改正？这种思路的程序是否值得再继续做下去？作为一个有价值的软件系统，程序要能够预见可能出现的错误，不能预见的错误要事后补救。程序员要知道如何避免和如何补救，不能补救的错误要及时放弃，采取别的策略。总之，无论采用什么方法都要把事情做成。

程序写完了，即使调通了，还必须再想想：程序还有哪些不足？还有哪些情况没有考虑到？是否还能进一步提高算法效率？要把事情做成，还要把事情做好，尽一切努力做得更好。不知不觉间，程序设计能力就会提高很多，“轻舟已过万重山”。

写程序是创作，创作过程是艰苦的，也是快乐的。当程序调试通过时，我们感受到成功的喜悦，哼着小曲，自鸣得意，心情畅快，“春风得意马蹄疾”。人的一生能有值得沉浸其中的事业是幸福的。

全书由叶核亚编著，南京大学计算机科学与技术系陈道蓄教授主审。

本书第 1 版于 2003 年出版，岁月如梭，转眼已十多年。感谢电子工业出版社十多年来对我的坚定支持；感谢陈老师认真细致地审阅全稿；感谢王少东、刘晓璐、徐金宝、彭焕峰、刘爱华、温志萍、程初老师和吴尚泽、郁中斐、吴腾阳等同学提供的帮助；感谢众多读者朋友的坚定支持以及提出的宝贵意见。大家对我的指导和帮助使我受益匪浅，受用终身。能与志同道合的人一起讨论共同关心的问题是愉快的，工作也因此变得更有动力。

对书中存在的不妥与错漏之处，敬请读者朋友批评指正。同时，呼吁每位读者购买正版图书，享受正版带来的有用的知识和应有的服务。

本书的全部例题和配套课件可从华信教育资源网站 (<http://www.hxedu.com.cn>) 下载，也可发邮件至 yeheya@x263.net 索取。

作者

目 录

第 1 章 Java 概述	1
1.1 了解 Java	1
1.1.1 Java 的诞生和发展	1
1.1.2 Java 的特点	2
1.1.3 Java 核心技术	5
1.2 JDK	7
1.2.1 JDK 的安装和设置	8
1.2.2 Application 应用程序的编辑、编译和运行	10
1.2.3 包	10
1.3 MyEclipse	15
1.3.1 MyEclipse 集成开发环境	15
1.3.2 Application 应用程序的编辑、编译和运行	16
1.3.3 设置 MyEclipse 环境属性	18
1.3.4 设置项目属性	20
1.3.5 程序调试技术	21
习题 1	23
实验 1 Application 应用程序的编辑、编译和运行	24
第 2 章 Java 语言基础	25
2.1 语言成分	25
2.1.1 标识符与关键字	25
2.1.2 基本数据类型	26
2.1.3 变量与常量	28
2.1.4 运算符与表达式	29
2.2 流程控制语句	35
2.2.1 顺序结构语句	36
2.2.2 选择语句	36
2.2.3 循环语句	38
2.2.4 转移语句	43
2.3 数组	44
2.3.1 一维数组	44
2.3.2 二维数组	47
2.4 静态方法	50
2.4.1 方法声明与调用	50

2.4.2	方法重载	51
2.4.3	参数传递	51
2.4.4	递归方法	56
2.5	字符串	57
	习题 2	62
	实验 2 Java 程序设计基础	63
第 3 章	类的封装、继承和多态	66
3.1	类和对象	66
3.1.1	类	66
3.1.2	对象	68
3.2	类的封装性	70
3.2.1	构造与析构	70
3.2.2	对象的引用和运算	72
3.2.3	访问控制	73
3.2.4	静态成员	75
3.2.5	浅拷贝与深拷贝	80
3.3	类的继承性	83
3.3.1	由继承派生类	83
3.3.2	继承原则及作用	84
3.3.3	子类的构造方法	86
3.4	类的多态性	87
3.4.1	子类声明多态成员	87
3.4.2	类型的多态	91
3.4.3	何时确定执行哪个多态方法? 怎样执行	91
3.4.4	多态的方法实现	93
3.5	类的抽象性	97
3.5.1	用继承刻画包含关系	97
3.5.2	抽象类	98
3.5.3	最终类	103
	习题 3	103
	实验 3 类的封装、继承和多态	105
第 4 章	接口、内部类和 Java API 基础	107
4.1	接口与实现接口的类	107
4.2	内部类和内部接口	113
4.3	Java API 基础	116
4.3.1	java.lang 包中的基础类库	116
4.3.2	java.util 包中的工具类库	121
4.4	泛型**	126

习题 4	129
实验 4 接口与实现接口的类.....	130
第 5 章 异常处理	132
5.1 异常处理基础	132
5.1.1 异常处理机制的必要性.....	132
5.1.2 错误	133
5.1.3 异常	134
5.2 异常处理措施	136
5.2.1 异常处理语句	137
5.2.2 抛出异常	139
5.2.3 定义异常类.....	140
习题 5	141
实验 5 异常的抛出、捕获并处理.....	142
第 6 章 图形用户界面设计	143
6.1 AWT 组件及属性类.....	143
6.1.1 AWT 组件.....	143
6.1.2 布局管理	147
6.1.3 颜色和字体.....	149
6.2 事件处理	149
6.2.1 委托事件模型	149
6.2.2 AWT 事件类和事件监听器接口	154
6.3 Swing 组件及事件	156
6.3.1 Swing 组件和布局	156
6.3.2 文本显示和编辑组件及事件	158
6.3.3 按钮组件	163
6.3.4 列表框和组合框组件及事件	164
6.3.5 中间容器	166
6.3.6 定制对话框.....	167
6.3.7 菜单组件	175
6.3.8 表格组件	183
6.4 图形设计	187
习题 6	193
实验 6 图形用户界面设计.....	195
第 7 章 多线程编程.....	203
7.1 进程和线程	203
7.1.1 进程	203
7.1.2 线程	205

7.2	Java 的线程对象	206
7.2.1	Runnable 接口和 Thread 类	207
7.2.2	线程对象的优先级	210
7.2.3	线程对象的生命周期	211
7.2.4	使用线程实现动画设计	212
7.3	线程互斥和线程同步	217
7.3.1	交互线程, 与时序有关的错误	217
7.3.2	交互线程的竞争与互斥	219
7.3.3	交互线程的协作与同步	223
	习题 7	234
	实验 7 线程设计	235
第 8 章	流和文件操作	238
8.1	文件和目录	238
8.1.1	文件	238
8.1.2	目录	240
8.1.3	文件系统	241
8.2	字节流	242
8.2.1	流的概念	242
8.2.2	抽象字节流	243
8.2.3	文件字节流	244
8.2.4	数据字节流	247
8.2.5	对象字节流	252
8.2.6	管道字节流	255
8.3	字符流	259
8.3.1	字符流类	259
8.3.2	Java 标准输入、输出	262
8.4	文件操作	263
8.4.1	文件类及其过滤器	263
8.4.2	文件选择对话框组件	267
	习题 8	276
	实验 8 流和文件操作	277
第 9 章	Socket 通信	278
9.1	网络编程基础	278
9.2	TCP Socket 通信	281
9.3	UDP Socket 通信	295
9.3.1	UDP Socket 点对点通信	295
9.3.2	UDP 组播通信	299
	习题 9	304

实验 9 Socket 通信.....	305
第 10 章 数据库应用.....	306
10.1 关系数据库系统.....	306
10.1.1 关系模型.....	306
10.1.2 客户-服务器结构的关系数据库系统.....	308
10.1.3 结构化查询语言 SQL.....	309
10.2 MySQL 数据库.....	313
10.2.1 安装数据库, 启动数据库服务.....	313
10.2.2 MySQL 数据库工作台.....	314
10.3 JDBC.....	316
10.3.1 JDBC 的作用和功能.....	316
10.3.2 指定 JDBC 驱动程序.....	317
10.3.3 连接数据库.....	318
10.3.4 执行 SQL 语句.....	319
10.3.5 处理数据查询结果集.....	320
习题 10.....	327
实验 10 数据库应用.....	327
第 11 章 基于 JSP 的 Web 应用.....	329
11.1 Web 浏览基础.....	329
11.2 JSP 技术.....	332
11.2.1 JSP 原理.....	332
11.2.2 运行 JSP.....	333
11.2.3 JSP 语法.....	338
11.2.4 基于 JSP 提供数据库应用服务的 Web 应用.....	345
习题 11.....	347
实验 11 基于 JSP 的 Web 应用设计.....	347
第 12 章 综合应用设计.....	349
12.1 集合框架.....	349
12.1.1 集合.....	350
12.1.2 映射.....	354
12.2 反射.....	355
12.3 使用复杂 Swing 组件.....	356
12.3.1 BorderLayout 盒式布局和 Box 容器.....	357
12.3.2 列表框.....	357
12.3.3 表格.....	362
12.3.4 多文档界面.....	362
12.3.5 树.....	367

12.4 数据库应用	383
12.5 课程设计的要求和选题.....	398
附录 A ASCII 字符与 Unicode 值.....	410
附录 B Java 语言的关键字	411
附录 C Java 基本数据类型.....	412
附录 D Java 语言的运算符及其优先级.....	413
附录 E java.lang 包 API (部分)	414
附录 F MyEclipse 常用菜单命令	419
参考文献	420

第 1 章 Java 概述

物竞天择，适者生存。同自然界的进化规律一样，程序设计语言、程序设计思想的变化和发展也是随着实际应用需要而变化 and 发展的。我们今天所看到、所使用的程序设计语言，经历了一系列竞争和淘汰之后仍然存在，说明它们有存在的道理，必定各有所长。这是一种自然选择的结果。

1995 年，Java 语言以一种具有跨平台特性、完全面向对象的程序设计语言问世，展现的是与众不同的全新的面貌，当年就获评十大优秀科技产品。之后凭借跨平台、健壮、安全、高效这些适应网络运行需要的特点，Java 快速成长，不但在 Internet 上游刃有余，而且通过 Java ME、Java SE、Java EE 三大平台，其应用领域全面覆盖嵌入式应用、桌面应用和企业级应用，所表现出的强大的应用系统设计能力，使 Java 无处不在。

本章简要介绍 Java 的特点和核心技术；介绍 Java Application 应用程序的基本形式，以及由虚拟机支持的运行机制；以 Windows 操作系统的 Java SE 版本为例，介绍 JDK 的安装和设置方法，以及编译、运行 Java 应用程序的方法；介绍在 MyEclipse 集成开发环境中编辑、编译和运行 Java 应用程序的方法，以及程序调试技术。

1.1 了解 Java

1.1.1 Java 的诞生和发展

1. 前身

1991 年，Sun 公司成立 Green 项目组，目的是开发嵌入家用电器的分布式软件系统，如交互式有线电视和家用电器的设备控制等，使电器更加智能化。由于这些电子设备品种繁多且标准各异，Green 项目组希望该控制系统具有简单、可靠、安全、容易联网和跨平台等特性，并且具有支持系统开发的编程工具。

Green 项目组最初采用 C++ 语言开发，由于 C++ 语言太复杂且安全性差，不能满足要求，于是 Green 项目组研究设计了一种新语言，取名为 Oak(橡树)，因为 Green 项目组负责人 James Gosling 办公室窗外有一棵大橡树。

Oak 语言保留了 C++ 语言的语法，为了简单、可靠、安全等特性，放弃了 C++ 语言的一些具有潜在危险特性的内容，如资源引用、指针、运算符重载等。Oak 具有的与平台无关的特性，使其适合网络编程。1994 年，Green 项目组用 Oak 编写的 Web 浏览器（称为 HotJava）展示了 Oak 作为 Internet 开发工具的能力。

2. 诞生

由于商标冲突，1995 年，Oak 语言更名为 Java 语言。Java 取名于印度尼西亚的爪哇岛，它盛产咖啡。Java 语言的标志就是一杯热咖啡。美国著名杂志《PC Magazine》将 Java 语言评

为 1995 年十大优秀科技产品。

Java 包括 Java 编程语言、开发工具和环境、Java 类库等。JDK (Java Development Kit, Java 开发工具包) 提供 Java 运行环境。1996 年, Sun 公司发布 JDK 1.0 和 HotJava。HotJava 通过嵌入在 Web 网页中的 Applet 运行 Java 程序, 一年之内, Microsoft、Netscape 等公司的 Web 浏览器宣布支持 Applet, 而 IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle 和 Microsoft 等公司相继购买 Java 技术许可证, 从此 Java 成为应用广泛的程序设计语言。

3. Java 2 平台

1998 年, Sun 公司发布 JDK 1.2, 即 Java 2 SDK (Software Development Kit)。得益于跨平台特性, Java 2 不仅能够应用于智能卡和小型消费类设备, 还能够应用于大型服务器系统; 提供的接口机制使得软件开发商、服务提供商和设备制造商能够紧密配合, 降低了软件开发和维护的工作量。

Sun 公司采取开放策略, 在其网站上可以免费获取 JDK, 这也是 Java 语言能够迅速发展的重要因素。不同的操作系统平台需要使用不同版本的 JDK。

4. Java ME、Java SE、Java EE 三大平台

1999 年, Sun 公司推出的 JDK 1.3 将 Java 平台划分为 J2ME、J2SE 和 J2EE, 这三个平台分别定位于嵌入式应用、桌面应用和企业级应用, 使 Java 技术获得了最广泛的应用。用户可根据实际应用领域的需求选择不同的 Java 平台。

2004 年, Sun 公司发布 J2SE 1.5, 自此 J2SE 1.5 更名为 J2SE 5.0。2005 年, Sun 公司发布 Java SE6, 并取消 Java 2 名称, Java 三大平台分别被更名为 Java ME、Java SE、Java EE。

- ❖ Java ME (Java Micro Edition) 是适用于小型设备和智能卡的 Java 嵌入式平台, 提供智能卡业务、移动通信、电视机顶盒等智能电器控制功能。
- ❖ Java SE (Java Standard Edition) 是适用于桌面系统的 Java 标准平台。Java SE SDK 也简称 JDK, 为创建和运行 Java 程序提供了最基本的环境, 包含 Java 编译器、Java 类库、Java 运行环境和 Java 命令行工具。
- ❖ Java EE (Java Enterprise Edition) 是 Java 的企业级应用平台, 提供分布式企业软件组件架构规范, 具有开放性、可扩展性、集成性和 Java EE 服务器之间的互操作性。

Web 应用是目前展示和操纵数据的主流技术。在 Web 应用开发技术中, Java EE 优势显著, 能够满足企业级应用对软件系统在功能、安全性、可靠性、高效性等方面的高要求, Java EE 已成为分布式企业级应用 (电子商务) 开发技术事实上的工业标准。

Oracle 公司于 2009 年收购了 Sun 公司, 于 2014 年发布 JDK 8, 于 2017 年 9 月发布 JDK 9。

1.1.2 Java 的特点

随着网络的飞速发展而发展, 作为软件开发的一种革命性技术, Java 的地位已被肯定。它在如此短暂的历史过程中, 经历如此规模的发展壮大, 显然并不是偶然的, 其有着内在的基础和外在的机遇。Java 语言建立在成熟的算法语言和坚实的面向对象理论的基础上, 具有强大的应用系统设计能力, 其具备的跨平台特性、面向对象和可靠性、安全性等特点是它能够充分适应网络需要的无可比拟的优势。Java 成为目前网络编程的首选语言, 充分说明了 Java 语

言的设计思想和其具有的特点适应了网络发展的特殊需要。不仅在网络应用方面，还在企业级应用领域，Java 以更简单、更精练的方式实现了 C++语言的所有功能。如今，Java 技术是当今世界信息技术的主流之一。

Java 应用如此广泛是因为其具有多方面的优势。

1. 跨平台特性

跨平台特性，也称为平台无关性，是指一个应用程序能够运行于不同的操作系统平台上，即 Sun 公司设计 Java 的宗旨“Write once, run anywhere”。跨平台特性使 Java 应用程序可以运行在多种操作系统（Windows、UNIX 等）平台上，这是 Java 区别其他高级语言的最重要标志。

Java 采用虚拟机技术支持跨平台特性。Java 虚拟机（Java Virtual Machine, JVM）是一套支持 Java 语言运行的软件系统，定义了指令集、寄存器集、类文件结构栈、垃圾收集堆、内存区域等，提供了跨平台能力的基础框架，如图 1-1 所示。Java 虚拟机运行于操作系统之上。

C/C++、Java 语言的运行方式比较如图 1-2 所示。

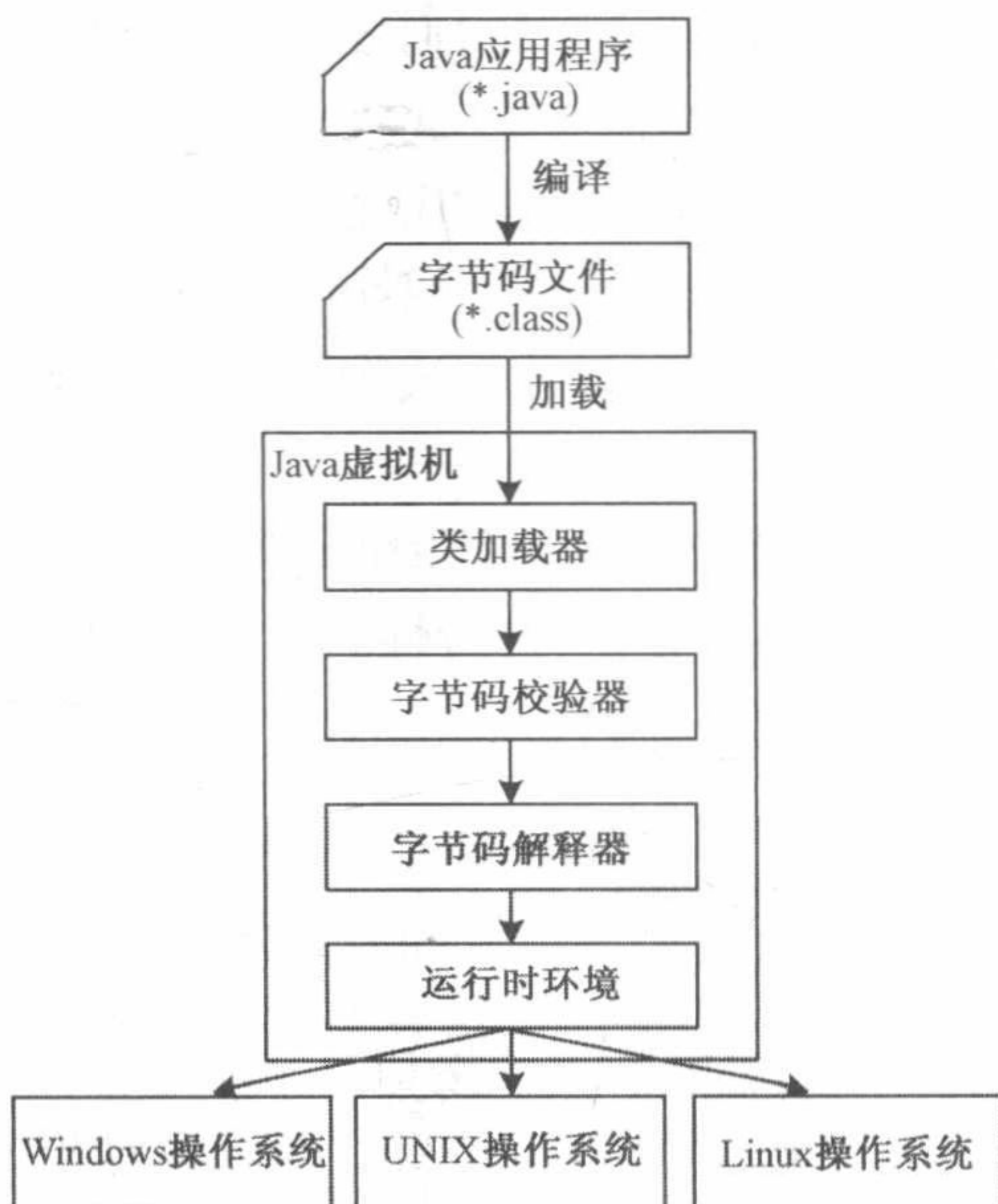


图 1-1 Java 虚拟机及其作用

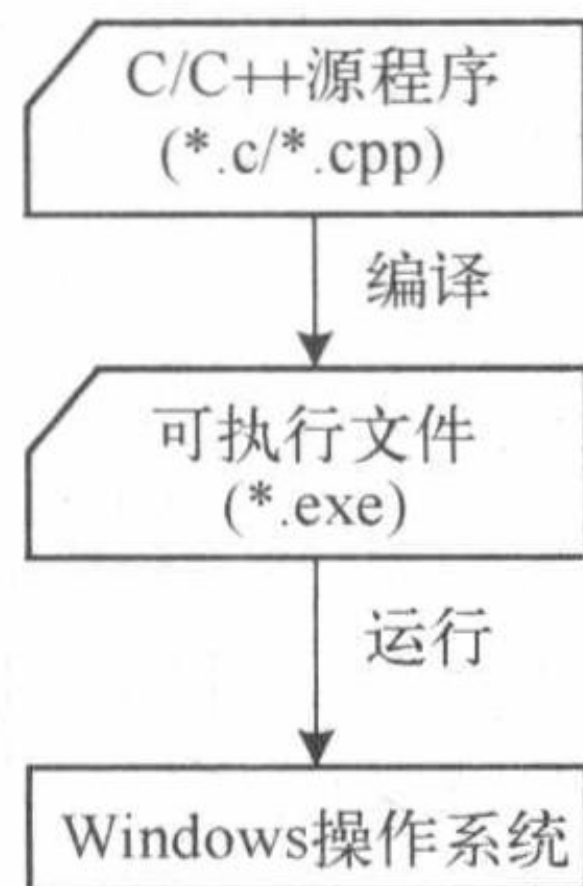


图 1-2 C/C++、Java 语言运行方式比较

Java 虚拟机是怎样运行 Java 程序的？首先来看 C/C++语言的运行方式，以此作比较。

在 Windows 操作系统中，C/C++等其他高级程序设计语言将源程序（*.c / *.cpp）编译生成可执行文件（*.exe，见图 1-2），再由 Windows 直接执行 EXE 文件。其他操作系统的可执行文件格式显然与此不同，如果程序要在不同操作系统之间运行，则必须移植。即使相同的操作系统，程序也会因不同的开发环境而不同。

Java 语言将源程序（*.java）编译生成字节码文件（*.class），也称为类文件，它是 Java 虚拟机的可执行文件格式，与各操作系统平台无关。

Java 虚拟机中的解释器负责解释执行字节码文件，将字节码解释成由本地操作系统支持的机器指令，解释一句，执行一句。因为每种操作系统都配有各自的 Java 虚拟机，所以这种运行方式使得一份 Java 源程序能够在不同操作系统上运行，“Write once, run anywhere”，实现了跨平台特性。

Java 虚拟机执行应用程序具有以下 3 个特点。

- ❖ 动态性：管理功能全部采用动态方式，如动态分配数组的存储空间、动态创建对象、动态连接数据库等，这些动态特性使 Java 程序适合在网络上运行。
- ❖ 异常处理：提供可靠的异常处理机制。
- ❖ 线程：采用多线程方式运行，各线程独立执行，并协调一致地处理共享数据。

Java 语言本身设计也体现出跨平台特性。例如，int 类型占用 4 字节（32 位），与操作系统是 16 位、32 位或 64 位无关。而 C 语言的 int 类型占用的字节数是可变的，与操作系统有关，16 位操作系统的 int 类型占用 2 字节，32 位操作系统的 int 类型占用 4 字节。

2. 完全面向对象和简单性

面向对象是当前软件开发的先进技术和重要方法。面向对象的概念是基于信息隐藏和数据抽象类型的概念，利用类和对象的机制将数据和方法封装在一起，通过统一的接口与外界交互；通过类的继承机制实现代码重用。面向对象方法反映了客观世界中现实的实体在程序中的独立性和继承性。这种方法有利于提高程序的可维护性和可重用性，还有利于提高软件开发效率和程序的可管理性。

Java 语言从 C++ 语言发展而来，有选择地继承了 C++ 语言的语法规则和面向对象的基本机制，放弃了 C++ 语言中一些含义模糊、过于复杂、安全性差、不适合网络应用的规则，但没有放弃与此相关的功能，采用更简单、功能更强、性能更好的方式实现 C++ 语言的所有功能。

Java 语言对 C/C++ 语言的基本语法改进说明如下。

- ① 不支持全局变量和宏替换，使用最终变量代替宏替换，避免全局变量和宏替换副作用。
- ② 为每种数据类型分配固定长度，实现数据类型的跨平台特性。
- ③ 进行类型相容性检查，防止不安全的类型转换。
- ④ 不支持 goto 语句。
- ⑤ 不支持指针类型，通过引用模型实现了指针的功能。
- ⑥ 不支持结构类型，使用类代替；不支持联合类型。
- ⑦ 不支持头文件，用 import 语句声明导入指定包中的类或接口。
- ⑧ 内存动态存储且自动管理，动态申请数组和对象的存储空间，自动释放空间，没有指针操作方式。

Java 语言是完全面向对象的，所有设计都必须在类中实现，一个 Java 应用程序就是多个类的集合。Java 语言对 C/C++ 语言的面向对象机制改进说明如下。

- ① 为 8 种基本数据类型提供相应的基本数据类型包装类，使基本数据类型与类相关联，体现完全面向对象。
- ② 将数组设计为引用类型，每个数组都有长度属性。
- ③ 不支持类似 C 语言那样的面向过程设计，不支持全局函数，所有函数都必须写在类中；函数参数不支持默认值形式，避免因默认值造成的二义性；函数内不能用 static 声明局部变量。
- ④ 不支持友元类和运算符重载，因为友元破坏封装性。
- ⑤ 提供单继承机制，即一个类只有一个父类，这样使得所有类（包括 Java 声明的类和程序员声明的类）能够形成具有树结构的类的层次体系，Java 为这个树结构设置了根类 Object。Object 类声明对象的基本状态和行为，这些行为可被所有对象继承。子类不能继承父类的构造

方法，但可以继承析构方法；所有成员方法都可在运行时被覆盖，都是 C++ 含义的虚函数；不支持多继承，提供接口，通过“单继承+接口”方式实现多继承功能。

Java 语言提倡简单性原则，对一个问题只提供一种简单、精练的表达方法，这样使程序简单、直接并且不造成歧义。例如，使用下标形式对数组元素进行操作，则不需要使用指针；方法（函数）采用返回值或引用类型参数返回结果，也不需要使用指针；有了类，则不需要结构类型；构造方法采用重载方式，则不需要采用参数默认值形式，避免产生歧义；通过成员方法实现类的操作，则不需要重载运算符等。因此，放弃结构、指针、多继承等，并没有影响 Java 语言的功能，Java 语言提供的机制具有更强的功能和更高的性能。

3. 可靠性

C++ 语言在稳定性和可靠性方面最大的隐患是使用指针和内存缺乏自动管理。Java 在语言和运行架构两个级别上提供程序运行稳定性和可靠性保证。

(1) 语言级别

Java 语言提供严密的语法规则，在编译和运行时进行严格检查，降低程序出错的可能性。例如，boolean 与 int 类型数据不能进行运算，数组下标不能越界，避免有效数据被覆盖，等等。

Java 语言提供异常处理机制，使程序具备在运行过程中及时发现并处理运行时错误的的能力，保证 Java 程序运行的稳定和可靠。

(2) 运行架构级别

Java 语言提供的资源回收（garbage collection）机制，对内存资源进行自动管理，跟踪程序使用的所有内存资源，自动收回不再被使用的内存资源。因此，程序中不需要写释放内存空间的语句。Java 自己操纵内存减少了内存出错的可能性，减轻了程序员的工作量，提高了程序运行的可靠性。

4. 安全性

Java 采用域管理方式的安全模型，无论是本地代码还是远程代码，都可以通过配置策略，设定可访问的资源域。这种策略使未经授权的代码不能对用户本地资源进行操作，更好地支持企业级应用，同时消除了区分本地代码和远程代码带来的困难。例如，Applet 应用程序在将远程 Web 页面下载到本地运行时，Java 会进行严格的代码安全性（code security）检测，限制许多可能危害网络安全的操作，如不能访问本地文件、不能建立新的网络连接等。

1.1.3 Java 核心技术

Java 的部分核心技术说明如下。

1. Application 应用程序

Java 应用程序有两种形式：Application 和 Applet。

Application 应用程序能够独立运行，有控制台和图形用户界面两种运行方式。

【例 1.1】接收命令行参数的 Application 应用程序。

本例演示基于控制台运行的 Application 应用程序，程序及说明如下。

```
// 控制台应用程序，功能是接收命令行参数作为输入数据，逐行输出；若无命令行参数，显示 Hello!  
public class Hello
```